Missingness Mechanisms

Ben

2025-02-05

In this document, I will go through a brief simulation demonstrating how to induce bias in CCA estimates for regression coefficients based on the results noted in (Oberman and Vink 2023, sec. 2.3; Buuren 2018, sec. 2.7 and Section 3.2.4). Please refer to these for further detail.

How to Get Biased Estimates with CCA

(Oberman and Vink 2023) notes that you don't always get biased estimates with CCA. In fact there are special cases where a seemingly MAR mechanism can function in practice as MCAR during simulations or where CCA is super-efficient while MI is biased under certain MNAR mechanisms. They discuss one condition in particular that is required for bias: the variable to be amputed must be correlated with the probability of being missing.

Other conditions and cases are discussed in (Buuren 2018, sec. 2.7).

- In single predictor regression of $Y = X\beta + \epsilon$, the CCA is equivalent to MI if only Y is incomplete when estimating regression coefficients.
  - If X is also incomplete or there are other variables to include in an imputation model, then MI is preferred.
- If missingness does not depend on Y, then the regression coefficients are unbiased under CCA with missing data on either (or both) X and Y.
- Logistic regression is unbiased under CCA with missing data on only Y or only X, but not both.

So to get a biased estimate with CCA we need:

1. An association between X and $Y; \beta \neq 0$.
2. Y must be associated with $P(R = 0)$ where $R = 0$ indicates missingness.
3. The to-be-amputed variable must be associated with $P(R = 0)$.

4. Missingness on the predictors X.
5. For MI to be more effective than CCA, more than Y must be related to the amputed variable.

So why didn't the original mechanism work?

The missingness mechanism as previously implemented failed point 3. of the above listed conditions. The association between the ILRs and stress response were also fairly weak. The way we generated the missing probabilities didn't ensure that there would be an association with the to-be-amputed variable and probabilities. Essentially generating the spurious MAR mentioned in (Oberman and Vink 2023) that behaves like MCAR.

Some Univariate Missingness Mechanisms

Consider the data set from the EMI survey. Let the stress EMI response be denoted Y and the ILR variables be denoted $X_1, \dots, X_5$ respectively. From the CCA analysis that we did to generate the simulated data, we know that $\beta_2 = 0.37$ corresponding to X_2 is the largest ILR regression coefficient. So since there's a known positive relationship between the stress response Y and X_2, I will focus our efforts on inducing missingness on X_2.

Three mechanisms inspired by (Buuren 2018, sec. 3.2.4) include:

1. MAR Right: $\text{logit}(P(R=0)) = -\alpha_0 + 3Y$
2. MAR Mid: $\text{logit}(P(R=0)) = -\alpha_0 - |Y - \tilde{Y}|$ where $\tilde{Y}$ is the median of Y.
3. MAR Tail: $\text{logit}(P(R=0)) = -\alpha_0 + |Y - \tilde{Y}|$

where α_0 is set in each case to guarantee the simulated missingness matches the desired proportion p_{miss}; $\alpha_0 = -\bar{U} - \log(1/p_{miss} - 1)$ and $\bar{U} = \frac{1}{n}\sum_{i=1}^{n} U_i$ and $U_i = 3Y_i$; $-|Y_i - \tilde{Y}|$; or $|Y_i - \tilde{Y}|$ respectively.

All we need is one of these mechanisms; so in the rest of this document we will use MAR Right, but the other two could work too.

On to the Simulation!

With these simulations, I am trying to induce bias in the CCA estimate of the regression coefficient corresponding to ILR2 when predicting the stress response. I will run these simulations over a range of missingness proportions in this document, but also evaluated other values of β and other mechanisms in a separate R script. I'm presenting these here because it minimizes the changes to the existing simulations; we keep β and the data generation the same, only the missingness mechanism changes slightly.

```
library(compositions, quietly = TRUE, warn.conflicts = FALSE)
library(mice, quietly = TRUE, warn.conflicts = FALSE)
library(zCompositions, quietly = TRUE, warn.conflicts = FALSE)
library(zoo, quietly = TRUE, warn.conflicts = FALSE)
library(dplyr, quietly = TRUE, warn.conflicts = FALSE)
library(generics, quietly = TRUE, warn.conflicts = FALSE)
library(readr, quietly = TRUE, warn.conflicts = FALSE)
library(ggplot2, quietly = TRUE, warn.conflicts = FALSE)

source("../idea_files_updated/impose_missing_corrected.R")
source("../idea_files_updated/generate_demographics.R")
```

```
condensed_data <- read.csv("../condensed_data.csv")
dat <- data.frame(condensed_data)
dat$physical <- rowMeans(dat[, c("EMI_2_2", "EMI_2_7", "EMI_2_8", "EMI_2_11", "EMI_2_16", "EM
dat$weight <- rowMeans(dat[, c("EMI_2_1", "EMI_2_4", "EMI_2_15", "EMI_2_18", "EMI_2_29", "EMI
dat$stress <-  rowMeans(dat[, c("EMI_2_3", "EMI_2_6", "EMI_2_20", "EMI_2_31", "EMI_2_34", "EM
dat$social <- rowMeans(dat[, c("EMI_2_5", "EMI_2_10", "EMI_2_19", "EMI_2_24", "EMI_2_33", "EM
dat$fitness <- rowMeans(dat[, c("EMI_2_13", "EMI_2_22", "EMI_2_27", "EMI_2_36", "EMI_2_41", "
dat$challenge <- rowMeans(dat[, c("EMI_2_9", "EMI_2_12", "EMI_2_14", "EMI_2_17", "EMI_2_23",

dat <- dat[dat$education != 1, ]

dat$cisgender <- ifelse(dat$Gender %in% c(0, 1), 1, 0)
```

Data Generation

In the simulations, I will use the data generating function implemented in the overall simulations. It generates demographics $X_{demo} = f_{demo}(E_{demo})$, then ILRs $X = f_{ILR}(X_{demo}, E_{ILR})$, and finally EMIs $Y = f_Y(X, X_{demo}, E_Y)$ where f_j is a function that converts E_j (random error for each component) and other variables into a new r.v. There are **no changes** at this step.

Missingness Mechanism

Here is the change!

We plug in the new alpha vector s.t. the mechanism is now MAR Right: $\text{logit}(P(R = 0)) = -\alpha_0 + 3Y$ where Y is the stress EMI response.

```r
sim_dat <- generate_demographics(dat = dat, beta = c(3.5, 0.1, 0.45, rep(0, 26)))
# cor(sim_dat$ilr2, sim_dat$stress)
sim_dat |>
  mutate(
    ilr2_p_miss1 = boot::inv.logit(1 - abs(stress - 5)),
    ilr2_p_miss2 = boot::inv.logit(-1 + abs(stress - 5)),
    ilr2_p_miss3 = boot::inv.logit(-5 + stress)

  ) |>
  ggplot(aes(stress, ilr2_p_miss3)) +
  geom_line() #+
```

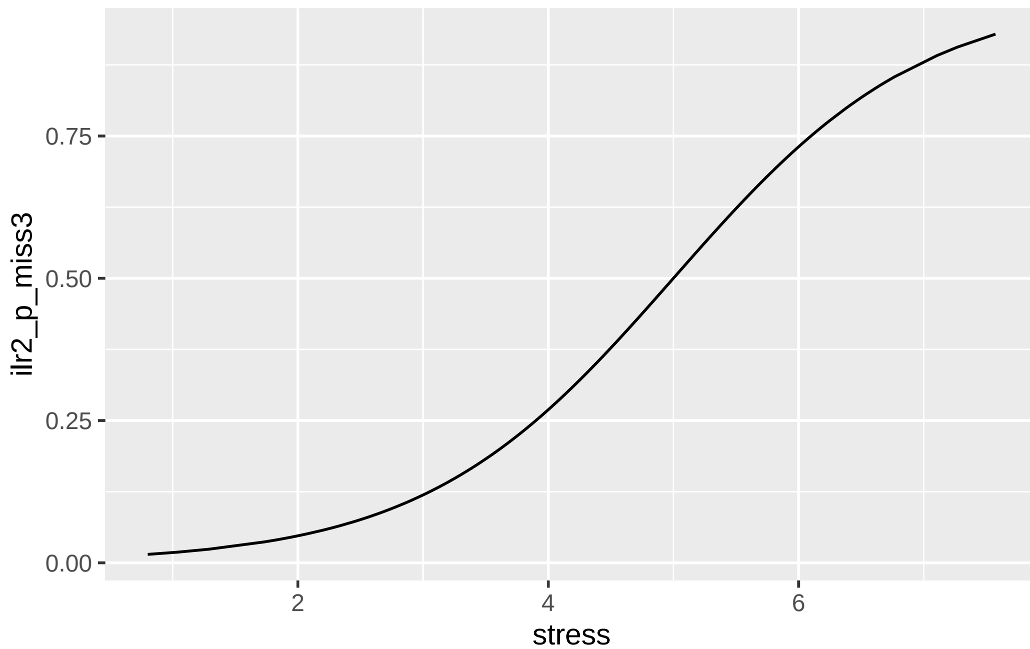


Figure 1: Missingness mechanism as function of stress.

```r
  # geom_line(aes(stress, ilr2_p_miss2), color = "darkred", linetype = "dotted") +
  # geom_line(aes(stress, ilr2_p_miss3), color = "navyblue", linetype = "dashed")
```

```r
plot_miss_mech <- function(sim_dat, p_miss = 0.5, mech = "MAR") {
  df_inc <- impose_missingness(sim_dat, p_miss = p_miss, mech = mech)
  df_inc$ilr2_inc <- df_inc$ilr2
  df_inc$ilr2 <- sim_dat$ilr2
  rho <- round(cor(df_inc$p_miss_2, df_inc$ilr2), 3)
```

```r
  p1 <- ggplot(df_inc, aes(ilr2, p_miss_2)) +
    geom_point() +
    geom_smooth(method = "lm") +
    labs(title = mech,
         subtitle = paste0("cor(ilr2, p_miss) = ", rho))

  p2 <- ggplot(df_inc, aes(ilr2, stress)) +
    geom_point(color = "dodgerblue") +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    geom_point(aes(ilr2_inc, stress), color = "darkred") +
    geom_smooth(aes(ilr2_inc, stress), method = "lm", color = "tomato", se = FALSE) +
    labs(title = mech,
         subtitle = paste0("cor(ilr2, p_miss) = ", rho))

  plt <- cowplot::plot_grid(p1, p2, nrow = 1)
  return(plt)
}

p1 <- plot_miss_mech(sim_dat, p_miss = 0.5, mech = "MAR")
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```

```r
p1
```

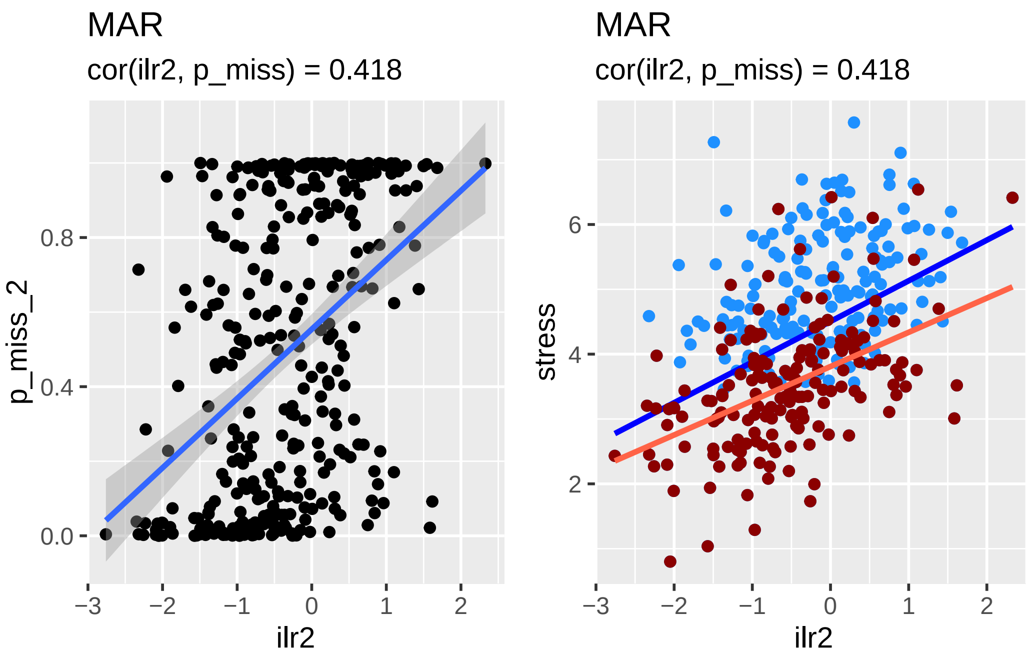


Figure 2: Associations between ILR2 and P(R=0) and between ILR2 and stress. Red is missing and blue is observed.

Analysis Model

The analysis model stays the same.

```
## Complete data
df_inc <- impose_missingness(sim_dat, p_miss = 0.5, mech = "MAR")
fit <- lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5 + Age + as.factor(income)
          + as.factor(gender) + as.factor(Ethnicity) + as.factor(education)
          + as.factor(Race), data = sim_dat)

smry1 <- summary(fit)
# smry1

## CCA
fit <- lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5 + Age + as.factor(income)
          + as.factor(gender) + as.factor(Ethnicity) + as.factor(education)
          + as.factor(Race), data = df_inc)
smry2 <- summary(fit)
# smry2
```

```r
## MI
imp <- mice::mice(df_inc, m = 3, method = "norm", printFlag = FALSE)
fit <- with(imp, lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5 + Age
                    + as.factor(income) + as.factor(gender)
                    + as.factor(Ethnicity) + as.factor(education)
                    + as.factor(Race)))
smry3 <- summary(pool(fit))

c("complete" = smry1$coefficients["ilr2", "Estimate"],
  "cca" = smry2$coefficients["ilr2", "Estimate"],
  "mi" = smry3[which(smry3$term == "ilr2"), "estimate"])
```

```
 complete       cca        mi
0.6227339 0.5221928 0.5297636
```

Simulation Structure

The simulation follows the same Monte Carlo set-up as usual.

1. Generate the data.
2. Impose missingness.
3. Fit the model using the complete data, CCA, and/or MI.
4. Collect results and format.

```r
fitted_cca_emi_model <- readRDS("~/Documents/GitHub/idea-grant-project/fitted_cca_emi_model.r
simulate_lm_cca <- function(sim_setting, N_sim, data, methods = c("cca")) {
  sim_res <- lapply(1:nrow(sim_setting), function(i) {
    print(sim_setting[i,])
    lapply(1:N_sim, function(q) {
      mech <- sim_setting$mech[i]
      beta0 <- 3.5
      beta_ilr2 <- sim_setting$beta[i]
      p_miss <- sim_setting$p_miss[i]
      beta_true <- coef(fitted_cca_emi_model)
      beta_true <- beta_true[1:29, "stress"]
      ## Generate data
      sim_dat <- generate_demographics(dat = data, beta = NULL)

      ## Impose Missingness
      df_inc <- impose_missingness(sim_dat, p_miss = p_miss, mech = mech)
      # Calculate association between stress and probability ILR2 is missing
```

```r
smry_mm <- summary(glm(R_2 ~ stress, data = df_inc,
                       family = binomial(link = "logit")))

z_mm <- smry_mm$coefficients[2, 3]

## Fit Complete Data Model
fit <- lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5 + Age
          + as.factor(income) + as.factor(gender)
          + as.factor(Ethnicity) + as.factor(education)
          + as.factor(Race),
          data = sim_dat)
smry <- summary(fit)
coef <- as.data.frame(smry$coefficients)
coef$Variable <- rownames(coef)
coef$true <- beta_true
rownames(coef) <- NULL
coef$method <- "complete"
for (method in methods) {

  if (method == "cca") {
    ## Fit the CCA Model
    tmp <- stats::na.omit(df_inc)
    fit <- lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5 + Age
              + as.factor(income) + as.factor(gender)
              + as.factor(Ethnicity) + as.factor(education)
              + as.factor(Race),
              data = tmp)
    smry <- summary(fit)
    coef2 <- as.data.frame(smry$coefficients)

    coef2$Variable <- rownames(coef2)
    rownames(coef2) <- NULL

    mean_stress_inc <- mean(tmp$stress)
  }
  else if (method == "mi") {
    ## Fit the MI model
    # print("running mi")
    tmp <- df_inc |>
      select(stress, ilr1, ilr2, ilr3, ilr4, ilr5,
             Age, income, gender, Ethnicity, education, Race)
```

```r
          # Impute 5 times (not enough, but a start)
          imp <- mice::mice(tmp, m = 5, method = "norm",
                            printFlag = FALSE, maxit = 5)
          fit <- with(imp, lm(stress ~ ilr1 + ilr2 + ilr3 + ilr4 + ilr5
                              + Age + as.factor(income) + as.factor(gender)
                              + as.factor(Ethnicity) + as.factor(education)
                              + as.factor(Race)))
          coef2 <- summary(pool(fit))
          coef2 <- coef2 |>
            mutate(
              Estimate = estimate,
              `Std. Error` = std.error,
              `t value` = statistic,
              `Pr(>|t|)` = p.value,
              Variable = term
            ) |>
            select(Estimate, `Std. Error`, `t value`, `Pr(>|t|)`, Variable)
        }
        coef2$true <- beta_true[1:nrow(coef2)]
        coef2$method <- method

        coef <- bind_rows(coef, coef2)
      }
      coef$iter <- q + (i - 1) * N_sim
      coef$beta1 <- beta_ilr2
      coef$z_stress_mm <- z_mm
      coef$p_miss <- p_miss
      coef$mech <- mech
      coef$diff_ilr2_mean <- mean(sim_dat$ilr2) - mean(df_inc$ilr2, na.rm = TRUE)
      coef$diff_stress_mean <- mean(sim_dat$stress) - mean_stress_inc
      coef$cor_pmiss_ilr2 <- cor(df_inc$p_miss_2, sim_dat$ilr2, use = "everything")
      return(coef)
    }) |>
      dplyr::bind_rows()
  }) |>
    dplyr::bind_rows()

  return(sim_res)
}
```

```r
sim_setting <- expand.grid(
  p_miss = seq(from = 0.25, to = 0.75, length.out = 7),
```

```r
  beta = 0.37,
  mech = c("MAR")
  # mech = c("MAR_MID", "MAR_TAIL", "MAR_RIGHT")
) |>
  mutate(
    # mech = "MAR"
  ) |>
  print()
```

```
     p_miss beta mech
1 0.2500000 0.37  MAR
2 0.3333333 0.37  MAR
3 0.4166667 0.37  MAR
4 0.5000000 0.37  MAR
5 0.5833333 0.37  MAR
6 0.6666667 0.37  MAR
7 0.7500000 0.37  MAR
```

```r
N_sim <- 250
start <- Sys.time()

sim_res <- simulate_lm_cca(sim_setting = sim_setting, N_sim = N_sim,
                           data = dat, methods = c("cca"))
```

```
  p_miss beta mech
1   0.25   -1  MAR
     p_miss beta mech
2 0.3333333   -1  MAR
     p_miss beta mech
3 0.4166667   -1  MAR
  p_miss beta mech
4    0.5   -1  MAR
     p_miss beta mech
5 0.5833333   -1  MAR
     p_miss beta mech
6 0.6666667   -1  MAR
  p_miss beta mech
7   0.75   -1  MAR
```

```
print(Sys.time() - start)
```

```
Time difference of 1.287331 mins
```

```
sim_res |>
  mutate(
    beta_label = stringr::str_c("beta_2 = ", round(beta1, 2))
  ) |>
  filter(stringr::str_detect(Variable, "ilr2")) |>
  ggplot(aes(as.factor(round(p_miss, 2)), cor_pmiss_ilr2)) +
  geom_boxplot() +
  facet_wrap(beta_label~Variable, ncol = 5)
```

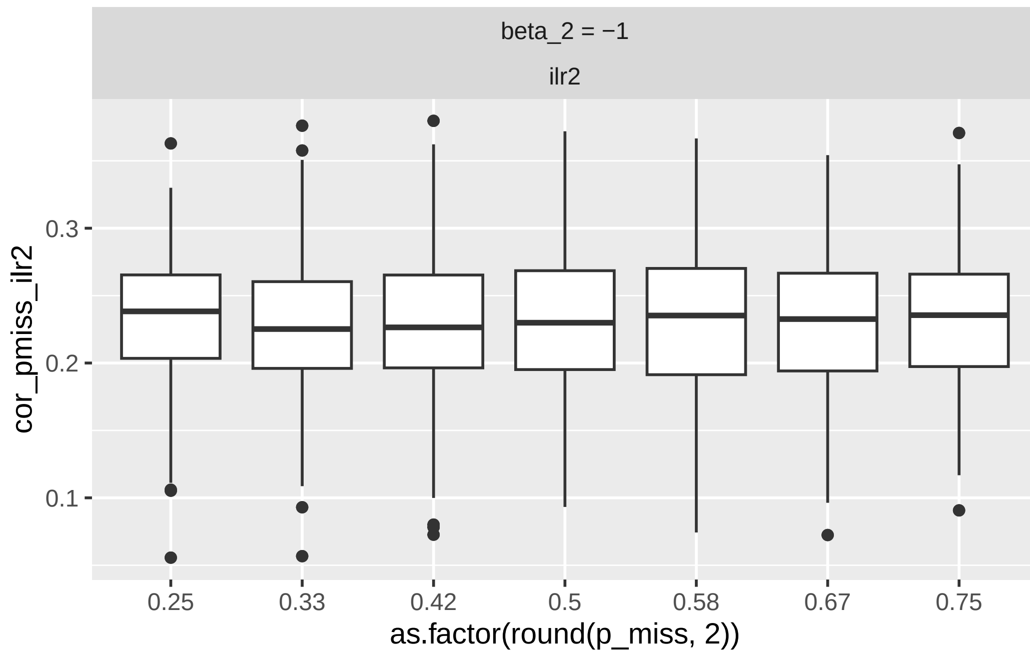


Figure 3: Boxplots of the correlations between ILR2 and P(R=0).

```
sum_sim <- sim_res |>
  mutate(
    beta_label = stringr::str_c("beta_2 = ", round(true, 2))
  ) |>
  filter(stringr::str_detect(Variable, "ilr2")) |>
  group_by(Variable, method, mech, beta_label, beta1, p_miss) |>
  summarize(
```

```
    n = n(),
    mean_est = mean(Estimate),
    sd_est = sd(Estimate),
    mean_bias = mean(Estimate - true),
    sd_bias = sd_est,
    mean_abs_bias = mean(abs(Estimate - true)),
    sd_abs_bias = sd(abs(Estimate - true)),
    mean_rel_bias = mean((Estimate - true) / abs(true)),
    mean_se = mean(`Std. Error`),
    sd_se = sd(`Std. Error`),
    mse = mean((Estimate - true)^2),
    sd_mse = sqrt(1/(N_sim - 1) * mean(((Estimate - true)^2 - mse)^2)),
    mean_cor = mean(cor_pmiss_ilr2),
    prop_sig_z_mm = mean(abs(z_stress_mm) > 1.96)
  ) |>
  ungroup()
```

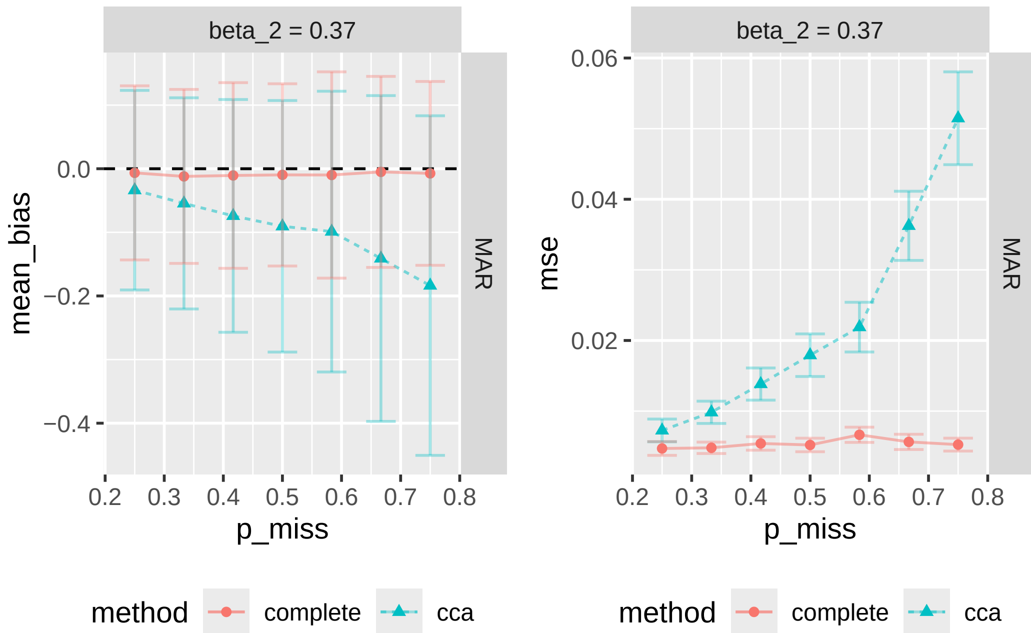
```
`summarise()` has grouped output by 'Variable', 'method', 'mech', 'beta_label',
'beta1'. You can override using the `.groups` argument.
```

```
sum_sim$beta_label <- forcats::fct_reorder(sum_sim$beta_label, sum_sim$beta1)
sum_sim$method <- forcats::fct_relevel(sum_sim$method, c("complete", "cca", "mi"))
```

```
Warning: 1 unknown level in `f`: mi
```

```
biasplt <- sum_sim |>
  ggplot(aes(p_miss, mean_bias, color = method)) +
  geom_hline(yintercept = 0, color = "black", linetype = "dashed") +
  geom_point(aes(shape = method)) +
  geom_line(aes(linetype = method), alpha = 0.5) +
  geom_errorbar(aes(x = p_miss,
                    ymin = mean_bias - 2 * sd_bias,
                    ymax = mean_bias + 2 * sd_bias,
                    color = method),
                width = 0.05, alpha = 0.35) +
  facet_grid(mech ~ beta_label, scales = "free") +
  theme(legend.position = "bottom")

absbiasplt <- sum_sim |>
  ggplot(aes(p_miss, mean_abs_bias, color = method)) +
```

```
  geom_hline(yintercept = 0, color = "black", linetype = "dashed") +
  geom_point(aes(shape = method)) +
  geom_line(aes(linetype = method), alpha = 0.5) +
  geom_errorbar(aes(x = p_miss,
                    ymin = mean_abs_bias - 2 * sd_abs_bias,
                    ymax = mean_abs_bias + 2 * sd_abs_bias,
                    color = method),
                width = 0.05, alpha = 0.35) +
  facet_grid(mech ~ beta_label, scales = "free") +
  theme(legend.position = "bottom")

relbiasplt <- sum_sim |>
  ggplot(aes(p_miss, mean_rel_bias, color = method)) +
  geom_hline(yintercept = 0, color = "black", linetype = "dashed") +
  geom_point(aes(shape = method)) +
  geom_line(aes(linetype = method), alpha = 0.5) +
  # geom_errorbar(aes(x = p_miss,
  #                   ymin = mean_abs_bias - 2 * sd_abs_bias,
  #                   ymax = mean_abs_bias + 2 * sd_abs_bias,
  #                   color = method),
  #               width = 0.05, alpha = 0.35) +
  facet_grid(mech ~ beta_label, scales = "free") +
  theme(legend.position = "bottom")

mseplt <- sum_sim |>
  ggplot(aes(p_miss, mse, color = method)) +
  geom_point(aes(shape = method)) +
  geom_line(aes(linetype = method), alpha = 0.5) +
  geom_errorbar(aes(x = p_miss,
                    ymin = mse - 2 * sd_mse,
                    ymax = mse + 2 * sd_mse,
                    color = method),
                width = 0.05, alpha = 0.35) +
  facet_grid(mech ~ beta_label, scales = "free") +
  theme(legend.position = "bottom")

cowplot::plot_grid(biasplt, mseplt, nrow = 1)
```




```r
# cowplot::plot_grid(biasplt, absbiasplt, relbiasplt, mseplt, nrow = 4)
```

```r
sum_sim |>
  select(method, n, beta1, mech, p_miss, mean_est, mean_bias, mse, mean_cor, prop_sig_z_mm)
  arrange(desc(abs(mean_bias)), desc(mse))
```

```
# A tibble: 14 x 10
   method       n beta1 mech  p_miss mean_est mean_bias     mse mean_cor
   <fct>    <int> <dbl> <fct>  <dbl>    <dbl>     <dbl>   <dbl>    <dbl>
 1 cca        250    -1 MAR    0.75     0.188  -0.184   0.0515     0.233
 2 cca        250    -1 MAR    0.667    0.231  -0.141   0.0362     0.229
 3 cca        250    -1 MAR    0.583    0.273  -0.0988  0.0219     0.232
 4 cca        250    -1 MAR    0.5      0.282  -0.0905  0.0179     0.231
 5 cca        250    -1 MAR    0.417    0.298  -0.0741  0.0138     0.230
 6 cca        250    -1 MAR    0.333    0.318  -0.0545  0.00983    0.229
 7 cca        250    -1 MAR    0.25     0.338  -0.0337  0.00726    0.234
 8 complete   250    -1 MAR    0.333    0.360  -0.0121  0.00480    0.229
 9 complete   250    -1 MAR    0.417    0.361  -0.0106  0.00542    0.230
10 complete   250    -1 MAR    0.583    0.362  -0.00984 0.00664    0.232
11 complete   250    -1 MAR    0.5      0.362  -0.00966 0.00520    0.231
12 complete   250    -1 MAR    0.75     0.365  -0.00733 0.00525    0.233
13 complete   250    -1 MAR    0.25     0.366  -0.00650 0.00471    0.234
14 complete   250    -1 MAR    0.667    0.367  -0.00499 0.00564    0.229
# i 1 more variable: prop_sig_z_mm <dbl>
```

Buuren, Stef van. 2018. *Flexible Imputation of Missing Data*. 2nd ed. Interdisciplinary Statistics Series. Chapman; Hall/CRC. https://stefvanbuuren.name/fimd/.

Oberman, Hanne I., and Gerko Vink. 2023. "Toward a Standardized Evaluation of Imputation Methodology." *Biometrical Journal* n/a (n/a): 2200107. https://doi.org/10.1002/bimj.202200107.